What is claimed is:

1. A system for pre-compiling a source cursor into a target library cache, comprising:

at least one source cursor stored in a source library cache, each source cursor comprising a statement with a shareable part and a non-shareable part;

an extraction process selectively copying the source cursor by extracting the shareable part of the statement from the source library cache; and

a compilation process pre-compiling the shareable part of the extracted source cursor into a target cursor without execution.

2. A system according to Claim 1, further comprising:

a lookup function creating a hash value from a text statement corresponding to the extracted source cursor, comparing the hash value to a set of target cursors stored in the target library cache and retrieving a reference pointer upon locating a matching target cursor.

3. A system according to Claim 2, further comprising:

a build function requesting a context area upon failing to locate a matching target cursor, loading the requested context area and building a new target cursor in the target library cache.

4. A system according to Claim 1, further comprising:

a parent cursor storing the target cursor as a parsed representation of a text statement corresponding to the extracted source cursor, the parent cursor comprising at least one child cursor.

5. A system according to Claim 4, further comprising:

for each session, the compilation process creating at least one child cursor for each text statement having identical text and different objects.

6. A system according to Claim 4, further comprising:

for each session, the compilation process creating at least one child cursor for each text statement having different session environments.

7. A system according to Claim 1, further comprising:

a target node asynchronously warming the target library cache prior to a switchover.

8. A system according to Claim 1, further comprising:

a target node asynchronously warming the target library cache prior to an unplanned failover.

9. A system according to Claim 1, wherein the extraction process extracts data selected from the group consisting of at least one of statement text, statement type, parsing user and parsing schema; parsing session environment; parsed representation and execution plan; and bind variable data.

10. A system according to Claim 1, wherein the extracted statement is written in a structured database language comprising at least one of SQL and PL/SQL.

11. A method for pre-compiling a source cursor into a target library cache, comprising:

storing at least one source cursor in a source library cache, each source cursor comprising a statement with a shareable part and a non-shareable part;

selectively copying the source cursor by extracting the shareable part of the statement from the source library cache; and

pre-compiling the shareable part of the extracted source cursor into a target cursor without execution.

12. A method according to Claim 11, further comprising:

creating a hash value from a text statement corresponding to the extracted source cursor;

comparing the hash value to a set of target cursors stored in the target library cache; and

retrieving a reference pointer upon locating a matching target cursor.

13. A method according to Claim 12, further comprising:

requesting a context area upon failing to locate a matching target cursor;

loading the requested context area; and

building a new target cursor in the target library cache.

14. A method according to Claim 11, further comprising:

storing the target cursor as a parsed representation of a text statement corresponding to the extracted source cursor, the target cursor comprising a parent cursor and at least one child cursor.

15. A method according to Claim 14, further comprising:

for each session, creating at least one child cursor for each text statement having identical text and different objects.

16. A method according to Claim 14, further comprising:

for each session, creating at least one child cursor for each text statement having different session environments.

17. A method according to Claim 11, further comprising:

asynchronously warming the target library cache prior to a switchover.

18. A method according to Claim 11, further comprising:

asynchronously warming the target library cache prior to an unplanned failover.

19. A method according to Claim 11, further comprising:

extracting data selected from the group consisting of at least one of statement text, statement type, parsing user and parsing schema; parsing session environment; parsed representation and execution plan; and bind variable data.

20. A method according to Claim 11, wherein the extracted statement is written in a structured database language comprising at least one of SQL and PL/SQL.

21. A computer-readable storage medium holding code for performing the method according to Claim 11.

22. A system for staging a pre-compiled cursor in a warmed instance cache, comprising:

a hash value created from a source cursor extracted from a source library cache, the source cursor comprising a shareable part and a non-shareable part;

a compilation process comparing the hash value to one or more target cursors maintained in a target library cache and retrieving a reference pointer to an address of a matching target cursor.

23. A system according to Claim 22, further comprising:

an open function opening a cursor definition entry in the target library cache.

24. A system according to Claim 23, further comprising:

a parse function instantiating the target cursor into the target library cache.

25. A system according to Claim 24, further comprising:

a bind function binding each input variable in the shareable part of the target cursor.

26. A system according to Claim 25, further comprising:

a describe function describing type definitions for each input variable in the target cursor without execution.

27. A system according to Claim 26, further comprising:

a close function closing the target cursor.

28. A method for staging a pre-compiled cursor in a warmed instance cache, comprising:

creating a hash value from a source cursor extracted from a source library cache, the source cursor comprising a shareable part and a non-shareable part;

comparing the hash value to one or more target cursors maintained in a target library cache; and

retrieving a reference pointer to an address of a matching target cursor.

29. A method according to Claim 28, further comprising:

opening a cursor definition entry in the target library cache.

30. A method according to Claim 29, further comprising:

instantiating the target cursor into the target library cache.

31. A method according to Claim 30, further comprising:

binding each input variable in the shareable part of the target cursor.

32. A method according to Claim 31, further comprising:

describing type definitions for each input variable in the target cursor without execution.

33. A method according to Claim 32, further comprising:

closing the target cursor.

34. A computer-readable storage medium holding code for performing the method according to Claim 28.

35. A method, comprising:

executing a database statement in a first database instance;

sending the database statement from the first database instance to a second database instance;

in the second database instance, generating and storing a structure required to prepare the database statement for execution in the second database instance;

receiving from a user or application a request to execute the database statement in the second database instance; and

after receiving the request, using the structure to execute the database statement in the second database instance.

36. A method according to Claim 35, wherein the structure is a parse tree for the database statement.

37. A method according to Claim 35, wherein the structure is an execution plan for the database statement

38. A method according to Claim 35, the sending operation occurs in anticipation of a planned shutdown of the first database instance.

39. A method according to Claim 35, wherein the sending operation occurs in anticipation of an unplanned shutdown of the first database instance that may possibly occur in the future.

40. A method, comprising:

receiving a database statement for execution in a first database instance;

generating in the first database instance, a structure required to prepare the database statement for execution;

executing the first database statement in the first database instance;

sending the structure from the first database instance to a second database instance;

receiving from a user or application a request to execute the database statement in the second database instance; and

after receiving the request, using the structure to execute the database statement in the second database instance.

41. A method according to Claim 40, wherein the structure is a parse tree for the database statement.

42. A method according to Claim 40, wherein the structure is an execution plan for the database statement.

43. A method according to Claim 40, the sending operation occurs in anticipation of a planned shutdown of the first database instance.

44. A method according to Claim 40, wherein the sending operation occurs in anticipation of an unplanned shutdown of the first database instance that may possibly occur in the future.